IN THE CLAIMS:

Please cancel Claims 1-23 and 30-32 without prejudice, and amend Claims 24 and 29 as follows:

24. (Once Amended) A method for programming a programmable system comprising a user programmable address arithmetic unit, the method comprising [the steps of]:

writing a first program in a first programming language, said first program configured to implement one or more user-defined address calculation functions in [an] said programmable address arithmetic unit; [and]

compiling said first program;

generating a first executable image, said first executable image adapted for loading into [an address arithmetic unit] a first program memory [of an] coupled to said programmable address arithmetic unit [in a processor having a programmable address arithmetic unit];

writing a second program in a second programming language, said second program configured to implement a desired [digital signal processing] algorithm;

compiling said second program into object code [for said processor], said object code comprising a plurality of machine level instructions for [said] a processor, said plurality of machine level instructions comprising at least one instruction [instructions] to control said programmable address arithmetic unit; and

generating a second executable image, said second executable image adapted for loading into a second program memory [of] coupled to said processor.

29. (Once Amended) The method of Claim 24, wherein said programmable address arithmetic unit [provides] comprises special purpose circuitry, said method further comprising [the step of]:

adapting said first program to use a software library to access said special purpose circuitry.

Also, please add Claims 33-76 as follows:

33. A method of supplying software, comprising:

supplying a first software module comprising instructions to implement an algorithm;

supplying a second software module comprising configuration codes to define the operation of a user-defined addressing mode in a programmable addressing arithmetic unit (PAAU);

whereby at least one instruction in said first module references an operand using said user-defined addressing mode.

34. The method of Claim 33, whereby at least some of said instructions in said first software module are used to program a digital signal processor.

35. The method of Claim 33, whereby said at least one instruction causes an auto-update to be applied to a pointer operand, the operation of the auto-update being defined at least in part by said user-defined addressing mode.

36. The method of Claim 35, further comprising specifying said auto-update using an assembly language mnemonic.

37. The method of Claim 36, whereby said first software module comprises a second instruction, said second instruction being used to specify one of a plurality of user-defined addressing modes to be selected to define, at least in part, the operation of said auto-update.

38. The method of Claim 33, further comprising configuring a programmable logic block within said PAAU using at least a portion of said configuration codes in said second software module.

39. The method of Claim 33, further comprising programming a set of sequential logic operations as implemented by a microsequenced state machine within said PAAU using at least a portion of said configuration codes in said second software module.

40. The method of Claim 33, further comprising programming a crossbar switching element within said PAAU using at least one of said configuration codes in said second software module.

41. The method of Claim 33, whereby said first software module comprises a plurality of subsets of instructions, each subset of instructions to be dispatched to one of a plurality of functional units in a multi-issue processor, the method further comprising dispatching at least one of said subsets to a functional unit comprising said PAAU.

42. A computer-readable medium containing a first software module having instructions defining the operation of an algorithm and a second software module containing configuration codes which define the operation of a user-defined addressing mode, said first and second modules implementing the method of:

executing said instructions in said first software module to implement said algorithm; and

using said configuration codes to configure the operation of said user-defined addressing mode;

whereby at least one instruction in said first module references an operand using said user-defined addressing mode.

43. The medium of Claim 42, whereby at least some of said instructions in said first software module are used to program a digital signal processor.

44. The medium of Claim 42, whereby said at least one instruction causes an auto-update to be applied to a pointer operand, and the operation of the auto-update is defined by said user-defined addressing mode.

45. The medium of Claim 44, whereby an assembly language mnemonic is used to specify said auto-update.

46. The medium of Claim 45, whereby said first software module comprises a second instruction, said second instruction specifying one of a plurality of user-defined addressing modes to be selected to define the operation of said auto-update.

47. The medium of Claim 42, whereby at least some of said configuration codes in said second software module are used to configure a programmable logic block within said PAAU.

48. The medium of Claim 42, whereby at least some of said configuration codes in said second software module are used to program a set of sequential logic operations as implemented by a microsequenced state machine within said PAAU.

49. The medium of Claim 42, whereby said configuration codes in said second software module are used to program a crossbar switching element within said PAAU.

50. The medium of Claim 42, whereby said first software module comprises a plurality of subsets of instructions, each subset of instructions to be dispatched to one of a plurality of functional units in a multi-issue processor, whereby one of said subsets is dispatched to a functional unit comprising said PAAU.

51. A method of executing software in a computerized system, said system comprising a processor with a programmable addressing unit (PAAU), a first software module containing instructions defining the operation of an algorithm and a second software module containing configuration codes defining the operation of a user-defined addressing mode supplied by said PAAU, the method comprising:

executing instructions in said first software module to implement said algorithm; and

using said configuration codes to configure the operation of said user-defined addressing mode;

whereby at least one instruction in said first module references an operand using said user-defined addressing mode.

52. The method of Claim 51, further comprising using at least some of said instructions in said first software module to program a digital signal processor.

53. The method of Claim 51, further comprising:

defining the operation of the auto-update by said user-defined addressing mode; and

causing an auto-update to be applied to a pointer operand using said at least one instruction.

54. The method of Claim 53, further comprising said auto-update using an assembly language mnemonic.

55. The method of Claim 54, whereby said first software module comprises a second instruction, and said method further comprises specifying one of a plurality of user-defined addressing modes to be selected to define the operation of said auto-update using said second instruction.

56. The method of Claim 51, further comprising using at least some of said configuration codes in said second software module to configure a programmable logic block within said PAAU.

57. The method of Claim 51, further comprising using at least some of said configuration codes in said second software module to program a set of sequential logic operations as implemented by a microsequenced state machine within said PAAU.

58. The method of Claim 51, further comprising using said configuration codes in said second software module to program a crossbar switching element within said PAAU.

59. The method of Claim 51, whereby said first software module comprises a plurality of subsets of instructions, each subset of instructions to be dispatched to one of a plurality of functional units in a multi-issue processor, the method further comprising dispatching at least one of said subsets to a functional unit comprising said PAAU.

60. A computerized system adapted for loading a first software module having a plurality of instructions and a second software module having at least one configuration code, the system comprising:

a processor having a programmable addressing arithmetic unit (PAAU);

whereby said processor is adapted to:

(i) execute at least a first one of said plurality of instructions to implement an algorithm,

(ii) configure a user-defined addressing mode in said PAAU using said at least one configuration code, and

(iii) execute at least a second one of said plurality of instructions, said at least second instruction referencing an operand using said user-defined addressing mode.

61. The system of Claim 60, whereby at least some of said instructions in said first software module are used to program a digital signal processor.

62. The system of Claim 60, whereby said at least one instruction causes an auto-update to be applied to a pointer operand, and the operation of the auto-update is defined by said user-defined addressing mode.

63. The system of Claim 62, whereby an assembly language mnemonic is used to specify said auto-update.

64. The system of Claim 63, whereby said first software module comprises a second instruction, said second instruction specifying one of a plurality of user-defined addressing modes to be selected to define the operation of said auto-update.

65. The system of Claim 60, whereby at least some of said configuration codes in said second software module are used to configure a programmable logic block within said PAAU.

66. The system of Claim 60, whereby at least some of said configuration codes in said second software module are used to program a set of sequential logic operations as implemented by a microsequenced state machine within said PAAU.

67. The system of Claim 60, whereby said configuration codes in said second software module are used to program a crossbar switching element within said PAAU.

68. The system of Claim 60, whereby said first software module comprises a plurality of subsets of instructions, each subset of instructions to be dispatched to one of a plurality of functional units in a multi-issue processor, whereby one of said subsets is dispatched to a functional unit comprising said PAAU.

69. A computer-implemented method for programming a processor comprising a programmable addressing arithmetic unit (PAAU), the method comprising:

allowing a sequence of instructions defining a program for implementing an algorithm to execute, thereby generating a sequence of addresses;

observing at least a subsequence of said sequence of addresses; and

generating a configuration program for said PAAU, said configuration program defining a user-defined addressing mode, whereby said user-defined addressing mode is capable of regenerating said subsequence.

70. The method of Claim 69, whereby said act of observing comprises observing a subsequence that corresponds to an address history sequence of a pointer variable.

71. The method of Claim 70, further comprising defining an auto-update operation using said user-defined addressing mode, said auto-update defining at least one method for advancing from a current address element of said subsequence to a successive address element of said subsequence.

72. The method of Claim 71, further comprising:

modifying said sequence of instructions by inserting into at least a subset of specified instructions a mnemonic that specifies said auto-update operation, such that when said modified sequence of instructions is executed, said pointer undergoes said observed subsequence of addresses.

73. The method of Claim 72, further comprising executing said sequence of instructions in N cycles and said modified set of instructions in M cycles, with the value of M being less than that of N.

74. A method of executing software in a computerized system, said system comprising a means for processing digital data, said means for processing comprising a programmable means for addressing, a first software module containing instructions defining the operation of an algorithm and a second software module containing configuration codes defining the operation of a user-defined addressing mode supplied by said means for addressing, the method comprising the steps of:

executing instructions in a first software module for implementing said algorithm; and

using said configuration codes for configure the operation of said user-defined addressing mode;

whereby at least one instruction in said first module is used for referencing an operand, said referencing of said operand being accomplished at least in part with said user-defined addressing mode.

75. A computerized system adapted for loading a first software module having a plurality of instruction means and a second software module having at least one configuration code means, the system comprising:

means for processing data having programmable addressing means;

whereby said means for processing is adapted to: